

The FLASH code (Fryxell et al. 2000) is a modular, component-based application code used for simulating compressible, reactive flows found in astrophysical environments. The code supports multiple methods for managing the discretized simulation mesh including the PARAMESH library (MacNeice. et al. 2000) for a block-structured adaptive mesh. The FLASH code traditionally scales very well (Calder et al. 2000) and was chosen in spring of 2004 as one of the marquee applications for the new BG/L platform. FLASH, with its adaptive mesh capability, was of particular interest to the developers at IBM because it was not immediately obvious that such codes would be appropriate for the machine. As one of the first applications to run, the FLASH code was instrumental in the early testing of the system software, and in fall 2005 the FLASH team was given the opportunity to perform a production science run on the LLNL BG/L. The run was performed on 65,536 processors in coprocessor mode over the period of approximately a week, for a combined total of almost 11 million CPU hours. The simulation chosen was one of compressible turbulence on an 1856^3 uniform mesh.

Our goal for the BGW run was to achieve a resolution study run on a 1024^3 mesh, filling in an important gap from lower resolution models achievable on the ANL BG/L, and the LLNL 1856^3 run. We had partial success; our run progressed to about a third of the total evolution time of the LLNL run, or about a half of the time to reach steady-state, which is the minimum desired evolution time. The reason for the limitation was partially due to the fact that we encountered an I/O bottleneck on BGW which we did not experience on the LLNL BG/L, on a problem of even greater scale. Thanks to follow-up work by Bob Walkup, this issue was identified as a metadata issue in our chosen file format. He has already outlined a possible solution, which involves a slightly different file format including an additional layer of subdirectories, to avoid metadata issues.

Using our previous build configurations on the LLNL BG/L, we found that porting the code and getting it to run on BGW took a little over an hour. The majority of this time was spent in successfully logging on remotely to BGW from ANL. We had ample access to IBM staff, including Ed Jedlicka and Tom Dirks at ANL and Jim Sexton at IBM, and with their assistance, we were able to log on. Later in the day, Bob Walkup aided us tremendously. It is quite fair to say that the assistance of these world-class people allowed us to make use of most of the time available to us, and (besides the raw computing power of BGW itself), were the strongest positives of BGW day.

There is room for improvement in the login process, and in access to BGW itself. On the first point, it may save both the IBM staff and users time if a brief login guide were provided in advance of BGW day, along with troubleshooting hints. On the second point, it would be useful to be able to access BGW more generally offsite. On the whole, though, our experience was tremendously positive, and we strongly believe that IBM is providing the computational sciences community a tremendous service by providing access to what is now the fastest open supercomputer in the world.

REFERENCES

- A. C. Calder, B. C. Curtis L. J. Dursi, B. Fryxell, G. Henry, P. MacNeice, K. Olson, P. Ricker, R. Rosner, F. X. Timmes, J. W. Truran, H. M. Tufo & M. Zingale, in Proc. Supercomputing 2000, IEEE Computer Soc. 2000, <http://sc2000.org> (Gordon Bell Prize)

B. Fryxell, et al. 2000, ApJS, 131, 273

P. MacNeice, K.M. Olson, C. Mobarry, R.de Fainchtein & C. Packer, Comp. Phys. Comm., 126, 330 (2000)